Technische Universität Ilmenau
Institut für Mathematik

Preprint No. M 14/06

The k-Server problem with parallel requests and the compound Harmonic algorithm

Regina Hildenbrandt

2014

Impressum:
Hrsg.: Leiter des Instituts für Mathematik
Weimarer Straße 25
98693 Ilmenau
Tel.: +49 3677 69-3621
Fax: +49 3677 69-3270
http://www.tu-ilmenau.de/math/

Preprint No. 14/6

The k-Server Problem with Parallel Requests and the Compound Harmonic Algorithm

R. Hildenbrandt, Department of Mathematics,
Ilmenau Technical University

1 Introduction

In [4] we have introduced a generalized k-server problem [1] with parallel requests where several servers can also be located on one point (which was initiated by an operations research problem). It is sensible in the case of parallel requests to distinguish the surplus-situation where the request can be completely fulfilled by means of the k servers and and the scarcity-situation where the request cannot be completely met.

By using a potential function we have shown that a corresponding Harmonic algorithm is competitive for this more general k-server problem against an adaptive online adversary in the case of unit distances. In this paper we investigate generalized k-server problems for general distances.

In the case of the scarcity-situation we will give an example for which the corresponding Harmonic algorithm is not competitive. In the other case, the surplus-situation, we will verify by an example that the potential function which was introduced by Y. Bartal and E. Grove (see [1], p. 6) is not helpful in order to prove competitiveness.

In Section 4 we will present the "compound Harmonic algorithm" for the generalized k-server problem in the case of the surplus-situation. Certain multi-step transition probabilities and absorbing probabilities are used by the compound Harmonic algorithm. For their computation one step of the generalized k-server problem is replaced by a number of steps of other

[1]The (usual) k-server problem was introduced by Manasse, McGeoch and Sleator [8]. Meanwhile it is the most studied problem in the area of competitve online problems. Historical notes on k-server problems can be found in the book by A. Borodin and R. El-Yaniv [2] (sections 10.9 and 11.7)[2] or also in the paper by Y. Bartal and E. Grove [1]. There the two important results are the competitiveness of the deterministic work-function algorithm (see E. Koutsoupias and C. Papadimitriou [5]) and of the randomized Harmonic k-server algorithm against an adaptive online adversary (see Y. Bartal and E. Grove [1]).

The work-function algorithm is an inefficient algorithm (with a good competitive ratio). In contrast the Harmonic k-server algorithm is memoryless and time-efficient. For this reason we first want to focus on a corresponding Harmonic k-server algorithm for the generalized k-server problem.

(generalized) specific k-server problems. We will show that this algorithm is competitive against an adaptive online adversary. The same bound of the competitive ratio as by Y. Bartal and E. Grove will be proved.

In the case of unit distances the Harmonic algorithm and the compound Harmonic algorithm are identical.

2 The formulation of the model [3]

[4] Let $k \geq 1$ be an integer, and $M = (M, d)$ be a finite metric space where M is a set of points with $|M| = N$. An algorithm controls k mobile servers, which are located on points of M. Several servers can be located on one point. The algorithm is presented with a sequence $\sigma = r^1, r^2, \cdots, r^n$ of requests where a request r is defined as an N-ary vector of integers with $r_i \in \{0, 1, \cdots, k\}$ ("parallel requests"). The request means that r_i servers are needed on point i $(i = 1, 2, \cdots, N)$. We say a request r is served if $\left\{ \begin{array}{c} \text{at least} \\ \text{at most} \end{array} \right\}$ r_i servers lie on i $(i = 1, 2, \cdots, N)$ in case $\left\{ \begin{array}{c} C[r,k] \\ C[k,r] \end{array} \right\}$. $C[r,k]$ denotes the case $\sum\limits_{i=1}^{N} r_i \leq k$ (surplus-situation, the request can be completely fulfilled) and $C[k,r]$ denotes the case $\sum\limits_{i=1}^{N} r_i \geq k$ (scarcity-situation, the request cannot be completely met, however it should be met as much as possible). By moving servers, the algorithm must serve the requests $r^1, r^2, \cdots, r^n$ sequentially. For any request sequence σ and any generalized k-server algorithm $ALG_{p(arallel)}$, $ALG_p(\sigma)$ is defined as the total distance (measured by the metric d) moved by the ALG_p's servers in servicing σ.

In this paper we will show that the corresponding compound Harmonic k-server algorithm is competitive (see Theorem 4.4) against an adaptive online adversary in the case of the scarcity-situation (for the definitions of competitive ratio and adaptive online adversary see [1] or [2], sections 4.1 and 7.1). Analogous to [2], p. 152 working with lazy algorithms ALG_p is sufficient. For that reason we define the set of feasible servers' positions with respect to the previous servers' positions s and the request r in the following way

$$\begin{aligned} &\hat{A}_{N;k}(s,r) \\ &\quad = \left\{ s' \in S_N(k) \;\middle|\; \begin{array}{l} r_i \leq s'_i \leq max\{s_i, r_i\},\ i = 1, \cdots, N,\ \text{in } C[r,k] \\ min\{s_i, r_i\} \leq s'_i \leq r_i,\ i = 1, \cdots, N,\ \text{in } C[k,r] \end{array} \right\} \end{aligned} \quad (1)$$

[3] See [4].

[4] For basic knowledge of (usual) k-server problems see also [2], chapters 10 and 11 for example.

$$\text{where } \; S_N(k) := \left\{ s \in \mathbb{Z}_+^N \mid \sum_{i=1}^{N} s_i = k \right\}. \tag{2}$$

The metric d implies that $(S_N(k), \hat{d})$ is also a finite metric space where $\hat{d}$ are the optimal values of the classical transportation problems with availabilities s and requirements s' of $S_N(k)$: $\sum_{i=1}^{N} \sum_{j=N}^{N} d(i,j)\, x_{ij} \;\to\; min$ subject to $\sum_{j=1}^{N} x_{ij} = s_i \;\forall i,\; \sum_{i=1}^{N} x_{ij} = s'_j \;\forall j,\; x \in \mathbb{Z}_+^N \times \mathbb{Z}_+^N$ (see [3], Lemma 3.6).

The corresponding Harmonic$_p$ k-server algorithm [5] operates as follows: Serves a (not completely covered) request r with randomly chosen servers so that for the (new) servers' positions $s' \in \hat{A}_{N;k}(s,r)$ is valid with respect to the previous servers' positions s and the request r. More precisely, Harmonic$_p$ leads to $s' \in \hat{A}_{N;k}(s,r)$ with probability

$$P_H(s'|s,r) = \frac{\frac{1}{\hat{d}(s,s')}}{\sum\limits_{s'':s'' \in \hat{A}_{N;k}(s,r)} \frac{1}{\hat{d}(s,s'')}}. \tag{3}$$

3 Considerations concerning the Harmonic$_p$ algorithm

At first we give an example that the Harmonic$_p$ algorithm is not competitive in general (if the case $C[k, r^t]$ is allowed).

Example 1 .
Let $k = 1$ and M be the set of integers with the usual metric.
If the server of the adversary is located on point $\bar{s} \in \mathbb{Z}$ and the server of the algorithm on point $s \in \mathbb{Z}$ then the adversary produces the requests r with

$$r_i = \begin{cases} \begin{cases} 1 & \text{if } \; i = s-1 \text{ or } \; i = s+1 \\ 0 & \text{otherwise} \end{cases} & \text{if } \; \bar{s} = s \\ \begin{cases} 1 & \text{if } \; i = \bar{s} \text{ or } \; i = s+1 \\ 0 & \text{otherwise} \end{cases} & \text{if } \; \bar{s} \neq s. \end{cases}$$

The adversary moves his server to another point (more precisely to $s-1$) if and only if the servers of the adversary and of the algorithm are located on the same point.

We assume that $\bar{s} \leq s$ at the beginning. Then

$$\bar{s} \leq s \tag{4}$$

[5] The Harmonic k-server algorithm has been introduced by P. Raghavan and M. Snir, see [8].

is valid in every step. Furthermore we will use the following symbols

$$\delta = \begin{cases} s - \bar{s} & if\, s \neq \bar{s} \\ 1 \; if \;\; s = \bar{s} \end{cases}$$

and

$h_\delta^l = E[cost(Harmonic_p\ algorithm)]$
with regard to l steps and δ at the beginning

$a_\delta^l = E[cost(adversary)]$
with regard to l steps and δ at the beginning

for the the expected costs. Then

$$\lim_{l\to\infty} \frac{h_1^l}{a_1^l} = \infty \tag{5}$$

can be proved (see Appendix A). Thus the $Harmonic_p$ algorithm is not competitive for such examples (with the severs' start position $s = \bar{s}$ and where the lengths l of the request sequences tends to infinity.)

For Example 2 and Section 4 we need the following definitions and statements. Y. Bartal and E. Grove have used a potential function Φ (for more information on this topic, see [1], Section 3) in order to show that the Harmonic k-server algorithm against an adaptive online adversary is competitive for the (usual) k-server problem. For further considerations let Φ_t denote the value of Φ at the end of the t-th step (corresponding to the t-th request r^t in the request sequence) and let $\Phi_t^\sim$ denote the value of Φ after the first stage of the t-th step (i.e., after the adversary's move and before the algorithm's move). If the potential function satisfies the following properties with regard to a randomized online algorithm Alg:

$$\Phi \geq 0 \tag{6}$$

$$\Phi_t^\sim - \Phi_{t-1} \leq C(k)D_t, \tag{7}$$

where D_t denotes the distance moved by the offline servers (controlled by the adversary) to serve the request in the t-th step,

$$E(\Phi_t^\sim - \Phi_t) \geq E(Z_t), \tag{8}$$

where Z_t represents the cost which incurred by the online algorithm to serve the request in the t-th step,

then algorithm Alg is C(k)-competitive (see [1], Lemma 1 and see also the following Lemma 4.2).

With regard to the Harmonic k-server algorithm Y. Bartal and E. Grove have constructed the following potential function: Let OFF be the set of offline servers, and ON the set of online servers. Y. Bartal and E. Grove have defined a weighted bipartite graph G on the online and offline servers in the following way (see [1], p. 6). Given an online server x and an offline server Y, then all paths from x to Y in $\{x\} \cup OFF$ are considered. The length of the j-th step of a path is weighted by a scaling function $\hat{f}_j$ [6] that is very large for small j and decreases monotonically. The weight of the edge from x to Y in G is the minimum scaled length of a simple path from x to Y in $\{x\} \cup OFF$. Let p be an assignment of servers to points in the metric space then

$$w(x, Y) = \min_{\{Y_1, \cdots, Y_l = Y\} \subset OFF} \left\{ \hat{f}_1 \cdot d\left(p(x), p(Y_1)\right) + \sum_{2 \le j \le l} \hat{f}_j \cdot d\left(p(Y_{j-1}), p(Y_j)\right) \right\}, \tag{9}$$

where $\hat{f}_j$ are weights with $\hat{f}_1 > \hat{f}_2 > \cdots > \hat{f}_N$.

The potential function is:

$$\Phi = \min_{\bar{M}\colon ON \leftrightarrow OFF} \sum_{x \in ON} w(x, \bar{M}(x)). \tag{10}$$

(This potential function is a function of the current locations of the online and offline servers. The weights $\hat{f}_j$ are computed in such a way that (8) is valid.)

Additionally, let

$\bar{s}$ ($\in S_N(k)$) denote the (offline) servers' positions controlled by the adversary at the end of the (t-1)-th step (i.e., at the beginning of the t-th step)

s ($\in S_N(k)$) denote the (online) servers' positions controlled by the algorithm at the beginning of the t-th step

s' ($\in \hat{A}_{N;k}(s, r^t)$) denote the (online) servers' positions at the end of the t-th step and

$\bar{s}'$ ($\in S_N(k)$) denote the (offline) servers' positions controlled by the adversary after the first stage of the t-th step.

The following example will show that the above-mentioned potential function is not helpful to to check wether the Harmonic$_p$ algorithm is competitive.

Example 2 .
Let $k = 4$ *and let a metric space* M *consist of 3 points* p_1, p_2, p_3 *with the pairwise distance of 1 and points* $p \in [2, \infty)$ *on the line. The distance of two*

[6] Y. Bartal and E. Grove have used the symbol $f(j)$.

points p_i, p_j $(i, j \notin \{1,2,3\})$ *on the line is* $d(p_i, p_j) = |p_i - p_j|$ *as usual. The distance* $d(p_i, p)$ *for* $i \in \{1,2,3\}$ *and* $p \in [2, \infty)$ *is defined as* $d(p_i, p) := p$.

At the beginning let the online and offline servers be located on $p_1, p_2, 2, 5$.

Additionally, we set:

$$d_0 = (d_0(l) =)\ 1 + l\ ,\ d_1 = (d_1(l) =) \begin{cases} 3 & \text{if}\, l = 1 \\ 3 + \frac{3}{2} \sum\limits_{q=3}^{l+1} \sqrt{q} & \text{if}\ \ l = 2, 3, \cdots \end{cases},$$

$$d_2 = (d_2(l) =)\ \tfrac{3}{2}\sqrt{l+2} \ \ \text{and}\ d_3 = (d_3(l) =)\ \tfrac{3}{2}\sqrt{l+1}\ \text{for}\ l = 1, 2, \cdots.$$

Possible server configurations $C_a(l), \cdots, C_i(l)$, *corresponding requests* r *and answers by the adversary can be found in Appendix B.*

Here we focus on the configuration $C_c(l)$:

$C_c(l)$: *ON is located on* $p_1, p_2, p_3,\ d_0 + d_1$.

OFF is located on $p_i, p_j,\ i, j \in \{1,2,3\},\ d_0, d_0 + d_1$.

r : *one server on* $d_0 + 1$ *and one server on* $d_0 + d_1 + d_2$

answer by the adversary: the offline servers on $d_0, d_0 + d_1$ *are moved.*

We will show that property (8) of the potential function for sufficiently large l *cannot be fulfilled.*

According to (9) and (10)
$\Phi_t^{\sim}(s, \bar{s}') = \hat{f}_1\ d_2 + \hat{f}_1 + \hat{f}_2 + \hat{f}_3\ (d_0 + 1)$ *for sufficiently large* l.

$\hat{A}_{N;k}(s, r)$ *includes the following 6 elements:*

$s'^{(1)}$: *servers on* $p_i, p_j,\ d_0 + 1, d_0 + d_1 + d_2$ *(that means* $s'^{(1)} = \bar{s}'$*)*
(where $Z_t(s, s'^{(1)}) = d_0 + 1 + d_2$*)*

$s'^{(2)}$: *servers on* $p_i, p_q,\ q \in \{1,2,3\} \setminus \{i, j\},\ d_0 + 1, d_0 + d_1 + d_2$
(where $Z_t(s, s'^{(2)}) = d_0 + 1 + d_2$*)*

$s'^{(3)}$: *servers on* $p_j, p_q,\ q \in \{1,2,3\} \setminus \{i, j\},\ d_0 + 1, d_0 + d_1 + d_2$
(where $Z_t(s, s'^{(3)}) = d_0 + 1 + d_2$*)*

$s'^{(4)}$: *servers on* $p_i,\ d_0 + 1, d_0 + d_1, d_0 + d_1 + d_2$
(where $Z_t(s, s'^{(4)}) = 2\ d_0 + 1 + d_1 + d_2$*)*

$s'^{(5)}$: *servers on* $p_j,\ d_0 + 1, d_0 + d_1, d_0 + d_1 + d_2$
(where $Z_t(s, s'^{(5)}) = 2\ d_0 + 1 + d_1 + d_2$*)*

$s'^{(6)}$: *servers on* $p_q,\ q \in \{1,2,3\} \setminus \{i, j\},\ d_0 + 1, d_0 + d_1, d_0 + d_1 + d_2$
(where $Z_t(s, s'^{(6)}) = 2\ d_0 + 1 + d_1 + d_2$*).*

The Harmonic$_p$ algorithm realizes $s'^{(i)}$ with probability:

$P_H(s'^{(i)}|s,r) = \frac{\frac{1}{Z_t(s,s'^{(i)})}}{N_f}$ *for* $i = 1, 2, \cdots 6$, *where* $N_f = \frac{3}{d_0+1+d_2} + \frac{3}{2\ d_0+1+d_1+d_2}$ *is referred to as the normalization factor.*

Using (9) and (10) computations yield

$\Phi_t(\bar{s}', s'^{(1)})\ P_H(s'^{(1)}|s,r)\ N_f = 0$

$\Phi_t(\bar{s}', s'^{(i)})\ P_H(s'^{(i)}|s,r)\ N_f = \frac{\hat{f}_1}{d_0+1+d_2} \quad (i = 2, 3)$

$\Phi_t(\bar{s}', s'^{(i)})\ P_H(s'^{(i)}|s,r)\ N_f = \frac{\hat{f}_1\ d_2+\hat{f}_2\ (d_1+d_2-1)+\hat{f}_3\ (d_0+1)}{2\ d_0+1+d_1+d_2} \quad (i = 4, 5)$

$\Phi_t(\bar{s}', s'^{(6)})\ P_H(s'^{(6)}|s,r)\ N_f = \frac{\hat{f}_1+\hat{f}_1\ d_2+\hat{f}_2\ (d_1+d_2-1)+\hat{f}_3\ (d_0+1)}{2\ d_0+1+d_1+d_2}$

for sufficiently large l.

Condition (8) is equivalent to $N_f\ \Phi_t^{\sim} - N_f\ E(\Phi_t) \geq 6$. This inequality can be written in the following representation:

$$\frac{\hat{f}_1\ d_2+\hat{f}_1+\hat{f}_2+\hat{f}_3\ (d_0+1)}{d_0+1+d_2} + 2\ \frac{\hat{f}_1\ d_2+\hat{f}_2+\hat{f}_3\ (d_0+1)}{d_0+1+d_2} + 2\ \frac{\hat{f}_1-\hat{f}_2\ (d_1+d_2-2)}{2\ d_0+1+d_1+d_2} + \frac{-\hat{f}_2\ (d_1+d_2-2)}{2\ d_0+1+d_1+d_2}$$

$$= \frac{3\ \hat{f}_1\ d_2+\hat{f}_1+3\ \hat{f}_2+3\hat{f}_3\ (d_0+1)}{d_0+1+d_2} + \frac{2\ \hat{f}_1-3\ \hat{f}_2\ (d_1+d_2-2)}{2\ d_0+1+d_1+d_2} \geq 6$$

If l tends to infinity then $3\ \hat{f}_3 - 3\ \hat{f}_2 \geq 6$ follows. This inequality is false since $\hat{f}_2 > \hat{f}_3$ is assumed.

Until now we do not have found a potential function in order to proof competitiveness and we do not know wether the Harmonic$_p$ algorithm in the case of the surplus-situation is competitive or not. [7] However, we will introduce the new "compound Harmonic$_p$ algorithm" in the following section and will prove the same bound of the competitive ratio as by Y. Bartal and E. Grove in the case of the surplus-situation.

4 The compound Harmonic$_p$ algorithm

In this section more than one server can be located on a point. For certain considerations we will also use the same potential function as by Y. Bartal and E. Grove (see (9) and (10)).

(Example: If we have three points p_1, p_2 and p_3 with the distances

[7] At least the Harmonic$_p$ algorithm is competitive if the number of points of the metric space is fixed.

$d(p_1, p_2) = 1, d(p_1, p_3) = d(p_2, p_3) = 5$ and the online servers' positions are given by $s = (3, 0, 0)$, the offline servers' positions by $\bar{s} = (0, 2, 1)$ then $\Phi(s, \bar{s}) = \hat{f}_1 + \hat{f}_1 + (\hat{f}_1 + 0 * \hat{f}_2 + 5 * \hat{f}_3)$ if $\hat{f}_3 < \frac{4}{5}\hat{f}_1$.)

It is simple to prove the following property of this potential function:

Lemma 4.1 .
If s_i (> 0) online severs and $\bar{s}_i$ (> 0) offline severs are located on point i then the number of $min\{s_i, \bar{s}_i\}$ online servers on point i are assigned to this number of offline severs on point i for the computation of the potential function $\Phi_t(s, \bar{s})$ by means of the minimum weight matching (see (9) and (10)).

At first we introduce following more specific

k-server problems where more than one server can be located on a point with a request where at most one server must be moved in servicing this request in a step and with additional blocking

(briefly speaking, k-server problems with blocking):

Besides creating the request the adversary can additionally block the same number of online and offline servers on points in a step (which are then not to be used in order to serve the request in this step). A request r with one $r_i > 0$ and $r_i = b_i + 1$, where b_i is the number of blocked online and offline servers, is allowed. Possibly that more than one server are located on a point.

(This also means that the probabilities which are implied by the HARMONIC$_p$ algorithm are different to those for models without blocking.)

Corresponding potential functions must be independent of blocking. Otherwise $\Phi_t \neq \Phi_{(t+1)-1}$ in general, where Φ_t is the potential function at the end of step t and $\Phi_{(t+1)-1}$ is the potential function at the beginning of step $t + 1$ (where some servers could be blocked).

A statement which is analogous with Lemma 1 by Y. Bartal and E. Grove (see [1]) is also valid:

Lemma 4.2 .
If there exists a potential function Φ, satisfying the properties (6), (7) and (8) with respect to some randomized online algorithm for the corresponding k-server problem with blocking (as above), then this algorithm is C(k)-competitive.

The proof is similar to the proof in [1]. Merely the random choices by the algorithm must also satisfy the conditions of blocking. [8]

Now we consider k-server problems with blocking (of the same number of online and offline servers) only on the point where the current request is placed by the adversary, (briefly speaking **k-server problems with blocking on the request point**).

Lemma 4.3 .
The Harmonic k-server algorithm related to the k-server problem with blocking on the request point is $((k+1)(2^k-1)-k)$*-competitive against an adaptive online adversary.*

The proof is similar to the proof in [1]. We use a corresponding potential function (see also the beginning of this section). If b_i online and offline servers are blocked on a point then these online and offline severs are assigned to each other for computation of the potential function by means of the minimum weight matching (which corresponds to Lemma 4.1). The values of the scaling function $\hat{f}$ and $C(k)$ can be computed analogous to the proof by Y. Bartal and E. Grove.

Lemma 4.4 .
The Harmonic k-server algorithm related to the k-server problem with blocking is competitive against an adaptive online adversary.

The proof of property (8) implies other inequalities for the computation of the values of the scaling function and yields another bound of the competitive ratio:

In *Case 1* (see [1], 5. Analysis of the Step-Change in the Potential Function, p. 6 - 8) a path $P(x) = \{X_1, \cdots, X_{l(x)}\}$ is considered. In relation to k-server problems with blocking it could be that several offline servers from the set $\{X_1, \cdots, X_{l(x)-1}\}$ lie on the same point.

Let l' be the number of blocked offline servers from the set $\{X_1, \cdots, X_{l(x)-1}\}$ $(l' \leq l(x)-1)$ and l'' be the number of the remaining blocked offline servers then $k' = k - l' - l''$ is the number of non-blocked offline (and online) servers.

By reason of blocked servers we must replace the last inequality $j\ f(j) \geq (k-j)\ f(j+1) + k$ on page 10 in [1] (which corresponds to $j\ \hat{f}_j \geq (k-j)\ \hat{f}_{j+1} + k$ using our symbols) by

$$(j-l')\ \hat{f}_j \geq (k'+l'-j)\ \hat{f}_{j+1} + k' = (k-l''-j)\ \hat{f}_{j+1} + k - l' - l''.$$

[8] Also in case of parallel requests such a lemma would be valid. However in order to use the lemma a corresponding potential function must be found.

Since $\hat{f}_j$ is increasing if l'' is decreasing we must consider the above inequality for $l'' = 0$ in order to compute a bound of the competitive ratio. $(j - l')\ \hat{f}_j \geq (k - j)\ \hat{f}_{j+1} + k - l'$ $(l' \leq j - 1)$ is equivalent to $\hat{f}_j \geq \frac{k-l'}{j-l'}\ (\hat{f}_{j+1} + 1) - \hat{f}_{j+1}$. If $l' = j - 1$ then $\frac{k-l'}{j-l'} = k - j + 1$ is the largest possible factor.

We set $\hat{f}_k = 1$ and then the other values of the scaling function can be successively calculated by means of the inequalities $\hat{f}_j \geq (k - j)(\hat{f}_{j+1} + 1) + 1$, $j = k - 1, \cdots, 1$. (Obviously, the values are greater than those in [1].) Finally, $C(k) = k\ \hat{f}_1 + (k - 1)\ \hat{f}_2$ follows as in [1] (see p. 9 and 10) by means of property (7).

Now we will consider a first subset of generalized k-server problems with parallel requests:

Generalized k-server problems where a multiple request on one point is allowed

The probabilities P_C which are implied by **the compound Harmonic$_p$ algorithm** and the proof of its competitiveness are derived from k-server problems with blocking on the request point.

Let s be the online servers' positions at the beginning of the t-th step and let $k > r_i > min\{1,\ s_i\}$ be a (multiple) request on point i (that also means $r_j = 0$ for $j \neq i$) in the t-th step.

Then we replace this t-th step of the generalized k-server problem by $r_i - s_i =: \bar{j}$ steps $t_1, t_2, \cdots, t_{\bar{j}}$ of a corresponding k-server problem with blocking on the request point i. More detailed that means the request r^j of the t_j-th step is $s_i + j$ on point i $(j \in \{1, 2, \cdots, \bar{j}\})$ and $s_i + j - 1$ online and offline servers on point i are blocked in this step.

If s' denotes the online servers' positions at the end of step t in relation to the generalized k-server problem then several sequences $(s'^1, s'^2, \cdots, s'^{\bar{j}})$ with $s'^j \in \hat{A}_{N;k}(s'^{j-1}, r^j)$ (where $s'^0 = s$) and $s'^{\bar{j}} = s'$ exist (in general) in relation to the corresponding k-server problem with blocking on the request point (where s'^j denotes the online servers' positions at the end of step t_j).

Now, the probabilities $P_C(s'|s, r)$ which will be defined for a multiple request on one point are $\bar{j}$-step transition probabilities. More detailed

$$P_C(s'|s, r) := \sum_{\{(s'^1, s'^2, \cdots, s'^{\bar{j}-1}, s')\}} P_H(s'^1|s, r^1) \cdot P_H(s'^2|s'^1, r^2) \cdot \cdots \cdot P_H(s'|s'^{\bar{j}-1}, r^{\bar{j}}) \tag{11}$$

where $P_H(s'^j|s'^{j-1}, r^j)$ $(j = 1, \cdots, \bar{j}, s'^0 = s, s'^{\bar{j}} = s')$ are be computed according to the Harmonic algorithm with the blocked servers in mind: If $s'^j_{l_0} = s'^{j-1}_{l_0} - 1$ then

$$P_H(s'^j | s'^{j-1}, r^j) = \frac{\frac{1}{d(l_0,i)}}{\sum\limits_{l: s'^{j-1}_l > 0, l \neq i} \frac{1}{d(l,i)}}. \tag{12}$$

Lemma 4.5 .
The compound Harmonic$_p$ algorithm related to the generalized k-server problem where a multiple request on one point is allowed is $((k+1)(2^k-1)-k)$*-competitive against an adaptive online adversary.*

Proof. We consider on the on hand the generalized k-server problem where a multiple request on one point is allowed and on the other hand a corresponding k-server problems with blocking on the request point as surrogate problem.

If the adversary moves the same servers in order to serve the requests in the $\bar{j}$ steps in relation to the surrogate problem (sp) as in servicing the multiple request in relation to the original problem (op) then the expected values $E_{op}[cost(adversary)(\sigma)]$ and $E_{sp}[cost(adversary)(\sigma')]$ are equal, where the request sequences σ' are constructed in relation to the original sequences σ as above.

$E_{op}[cost(compound\ Harmonic_p\ algorithm)(\sigma)]$ and
$E_{sp}[cost(compound\ Harmonic_p\ algorithm)(\sigma')]$ are also equal because of (11).

Then the statement follows by means of Lemma 4.3. ■

k-server problems with parallel requests

k-server problems where a multiple request on one point is allowed are used for computation of the probabilities P_C which are implied by **the compound Harmonic$_p$ algorithm** related to k-server problems with (proper) parallel requests.

Let s denote the online servers' positions at the beginning of the t-th step and let r be the request in the t-th step with (w.l.o.g.)

$$\begin{cases} r_i > 0 & \text{for} \quad i = 1, \cdots, \bar{N} \\ r_i = 0 & \text{for} \quad i = \bar{N}+1, \bar{N}+2, \cdots, N \end{cases}, \ \bar{N} \leq N \text{ and } \sum_{i=1}^{N} r_i < k.$$

Then we replace this t-th step of the generalized k-server problem with (proper) parallel requests by a number of steps of a corresponding k-server problem where a multiple request on one point is allowed. More detailed the following request sequence $(\bar{r}^j)_{j=1,2,\cdots}$ (with a multiple request on one point per step) for the surrogate steps should be created by the adversary:

$$\bar{r}^j_i = \begin{cases} r_i \text{ if } j \equiv i \ mod(\bar{N}) \\ 0 \text{ otherwise} \end{cases}.$$

If s' denotes the online servers' positions at the end of step t in relation to the generalized k-server problem then several sequences $(s'^1, s'^2, \cdots, s'^{\bar{j}})$ with several length $\bar{j}$, $s'^j \in \hat{A}_{N;k}(s'^{j-1}, \bar{r}^j)$ (where $s'^0 = s$) and $s'^{\bar{j}} = s'$ exist (in general) in relation to the corresponding k-server problem where a multiple request on one point is allowed (where s'^j denotes the online servers' positions at the end of step t_j). If $s'^j \geq \bar{r}_i^{j+1} > 0$ then the corresponding surrogate step could be also omitted.

Such sequences represent realizations of a time-homogeneous Markov chain (see [6] for example) with the absorbing state s' and transition probabilities $P_C(s'^j|s'^{j-1}, \bar{r}^j)$ related to a multiple request on one point as above.

The probabilities $P_C(s'|s, r)$, $s' \in \hat{A}_{N;k}(s, r)$ for (proper) parallel requests, which are used by the compound Harmonic$_p$ algorithm, are defined as absorbing probabilities. Absorbing probabilities can be computed by means of linear systems (see [6], Theorem 6.6 and the following Example 3). For this purpose all states of the above mentioned Markov chains must be known and the corresponding transition probabilities are the coefficients of these linear systems. The number of these states is finite.

Furthermore

$$\sum_{s' \in \hat{A}_{N;k}(s,r)} P_C(s'|s, r) = 1 \tag{13}$$

is valid and the solutions of the linear systems are unique.

Theorem 4.6 .
The compound Harmonic$_p$ algorithm related to the generalized k-server problems with parallel requests is $((k+1)(2^k - 1) - k)$-competitive against an adaptive online adversary in the case of the surplus-situation.

Proof. We consider on the on hand the generalized k-server problems with parallel requests and on the other hand corresponding k-server problems where a multiple request on one point is allowed.

If the adversary moves the same servers in order to serve the requests in the first $\bar{N}$ steps in relation to the surrogate problem (sp) as in servicing the parallel request in relation to the original problem (op) then the expected values $E_{op}[cost(adversary)(\sigma)]$ and $E_{sp}[cost(adversary)(\sigma')]$ are equal, where the request sequences σ' are constructed in relation to the original sequences σ as above. We can use such surrogate sequences since (13) is valid.

By means of the triangle-inequality
$E_{op}[cost(compound\ Harmonic_p\ algorithm)(\sigma)] \leq$
$E_{sp}[cost(compound\ Harmonic_p\ algorithm)(\sigma')]$ follows.

Then the application of Lemma 4.5 leads to the bound of the competitive ratio. ∎

Corollary 4.7 .
The compound Harmonic$_p$ algorithm related to the generalized k-server problems with parallel requests $(r^1, r^2, \cdots)$, where $r_i^j \leq 1$ for any i and j and $\sum_{i=1}^{N} r_i^j \leq k$ for any j, have the same competitive ratio as the Harmonic algorithm related to the (usual) k-server problems against an adaptive online adversary.

Proof. Because of $r_i^j \leq 1$ for any i and j, sequences σ' present request sequences for usual k-server problems and

$$E_{sp}[cost(compound\ Harmonic_p\ algorithm)(\sigma')]$$
$$= E_{sp}[cost(Harmonic\ algorithm)(\sigma')].$$ ■

Example 3 .
Let $k = 4$ and let the metric space M consist of 6 points $p_1, p_2, \cdots, p_6$ of the two-dimensional Euclidean space with the distances $d(p_3, p_1) = 5$, $d(p_4, p_1) = 3,85, d(p_5, p_1) = 1,6, d(p_6, p_1) = 4,5, d(p_2, p_1) = 2,4$ and $d(p_3, p_2) = 4, d(p_4, p_2) = 5, d(p_5, p_2) = 2,1, d(p_6, p_2) = 4,55$.

The current online servers' positions are given by $s = (0,0,1,1,1,1)^T$ and the current requests by $r = (1,1,0,0,0,0)^T$.

Then we have 6 feasible online servers' positions with respect to s and r:

$$s'^{(1)} = (1,1,0,0,1,1)^T, s'^{(2)} = (1,1,0,1,0,1)^T, s'^{(3)} = (1,1,0,1,1,0)^T,$$
$$s'^{(4)} = (1,1,1,0,0,1)^T, s'^{(5)} = (1,1,1,0,1,0)^T, s'^{(6)} = (1,1,1,1,0,0)^T.$$

Corresponding distances $\hat{d}(s, s'^{(i)})$, probabilities $P_H(s'^{(i)}|s,r)$ (according to the Harmonic$_p$ algorithm) and $P_C(s'^{(i)}|s,r)$ (according to the compound Harmonic$_p$ algorithm) can be found in the following table

i	1	2	3	4	5	6
$\hat{d}(s, s'^{(i)})$	7,85	5,60	8,50	2,95	8,40	6,15
$P_H(s'^{(i)}\|s,r)$	0,1459	0,2045	0,1347	0,1924	0,1363	0,1862
$P_C(s'^{(i)}\|s,r)$	0,0836	0,2504	0,0781	0,2582	0,0829	0,2466

.

$P_H(s'^{(i)}|s,r)$ can be calculated according to (3) and $\hat{d}(s, s'^{(i)})$ by means of the classical transportation problem, see Section 2. For the computation of $P_C(s'^{(i)}|s,r)$ see Appendix C.
(We can observe that $P_C(s'^{(i)}|s,r) < P_H(s'^{(i)}|s,r)$ for greater distances $\hat{d}(s, s'^{(i)})$ and $P_C(s'^{(i)}|s,r) > P_H(s'^{(i)}|s,r)$ for smaller $\hat{d}(s, s'^{(i)})$.)

Remarks 1 .

(i) In the case of unit distances (that means $d(i,j) = 1\ \forall\ i \neq j$) all probabilities $P_C(s'|s,r)$ are the same for $s' \in \hat{A}_{N;k}(s,r)$.
Hence $P_C(s'|s,r) = P_H(s'|s,r)$ and the compound Harmonic$_p$ algorithm and the Harmonic$_p$ algorithm are identical.

(ii) For k-server problems with (proper) parallel requests we could also introduce another compound Harmonic$_p$ algorithm, where $\sum_{i=1}^{\bar{N}}(r_i - s_i)$-step transition probabilities would be used instead of the absorbing probabilities and where several servers on several points must be blocked. Then Lemma 4.4 would imply the competitiveness of such an algorithm, however with a weaker bound of the competitive ratio.

References

[1] Y. Bartal and E. Grove, The Harmonic k-Server Algorithm Is Competitive. *Journal of the ACM*, Vol. 47, No. 1 (2000) pp. 1 - 15.

[2] A. Borodin and R. El-Yaniv, *Online computation and competitive analysis*, University Press, Cambrigde 1998.

[3] R. Hildenbrandt, *Methoden aus ganzzahliger Optimierung und Verbandstheorie zur Behandlung eines stochastischen dynamischen Transportproblems*, Habilitationsschrift, TU Ilmenau 1995, (Libri BoD 2000).

[4] R. Hildenbrandt, A k-server problem with parallel requests and unit distances *Information Processing Letters* 114/5 (2014) pp. 239-246 DOI information: 10.1016/j.ipl.2013.12.011

[5] E. Koutsoupias and C. Papadimitriou, On the k-server conjecture, *Journal of the ACM*, 42(5) (1995) pp. 971-983.

[6] P. Langrock and W. Jahn, *Einführung in die Theorie der Markovschen Ketten und ihre Anwendungen*, Teubner, Leipzig 1979.

[7] M.S. Manasse, L.A. McGeoch and D.D. Sleator, *Competitive algorithms for on-line problems*, In Proceeding of the 20th Annual ACM Symposium on Theory of Computing (1988) pp. 322 - 333 (Journal version).

[8] P. Raghavan and M. Snir, *Memory versus randomization in on-line algorithms*, In Proceeding of the 16th ICALB. Vol. 372 of Lecture Notes in Computer Sciences (1989) pp. 687 - 703.

[9] D. D. Sleator and R. E. Tarjan, Amortized efficiency of list update and paging rules, *Communications of the ACM* 28, No. 2 (1985) pp. 202 - 208.

Appendix

A Proof of (5) concerning Example 1

According to the Harmonic$_p$ algorithm

$$h_\delta^1 = \frac{1}{\delta+1} \cdot \delta + \frac{\delta}{\delta+1} \cdot 1 = 2 \cdot \frac{\delta}{\delta+1} \quad \text{for} \quad \delta = 1, 2, ... \tag{14}$$

$$\text{and} \qquad a_1^1 = 1, \ a_\delta^1 = 0 \ \text{ for } \delta = 2, 3, ... \tag{15}$$

follow. Using (14) and (15) recursion computations lead to

$$h_\delta^{l+1} = \tfrac{1}{\delta+1}(\delta + h_1^l) + \tfrac{\delta}{\delta+1}(1 + h_{\delta+1}^l) = \tfrac{1}{\delta+1}(h_1^l + \delta h_{\delta+1}^l) + h_\delta^1 \tag{16}$$

$$a_\delta^{l+1} = \qquad \tfrac{1}{\delta+1}(a_1^l + \delta a_{\delta+1}^l) + a_\delta^1 \tag{17}$$

$$\text{for } \delta = 1, 2, ...$$

At first we consider more general sequences $(g_\delta^l)_{l=1,2,...}$ with
$g_\delta^{l+1} = \frac{1}{\delta+1}(g_1^l + \delta g_{\delta+1}^l) + g_\delta^1$ and any given g_δ^1 for $\delta = 1, 2, ...$

We will show by mathematical induction that

$$\begin{aligned}
g_\delta^l = {} & g_1^1 \left[\tfrac{1}{\delta+1} a_1^{l-1} + \tfrac{\delta}{(\delta+1)(\delta+2)} \cdot a_1^{l-2} + \tfrac{\delta}{(\delta+2)(\delta+3)} \cdot a_1^{l-3} + \cdots + \tfrac{\delta}{(\delta+l-2)(\delta+l-1)} a_1^1\right] \\
& + \ g_2^1 \cdot \tfrac{1}{2} \left[\tfrac{1}{\delta+1} a_1^{l-2} + \tfrac{\delta}{(\delta+1)(\delta+2)} a_1^{l-3} + \tfrac{\delta}{(\delta+2)(\delta+3)} a_1^{l-4} + \cdots + \tfrac{\delta}{(\delta+l-3)(\delta+l-2)} a_1^1\right] \\
& + \ g_3^1 \cdot \tfrac{1}{3} \left[\tfrac{1}{\delta+1} a_1^{l-3} + \tfrac{\delta}{(\delta+1)(\delta+2)} a_1^{l-4} + \cdots + \tfrac{\delta}{(\delta+l-4)(\delta+l-3)} a_1^1\right] \\
& + \\
& \vdots \\
& + \ g_{l-2}^1 \cdot \tfrac{1}{l-2} \left[\tfrac{1}{\delta+1} a_1^2 + \tfrac{\delta}{(\delta+1)(\delta+2)} a_1^1\right] \\
& + \ g_{l-1}^1 \cdot \tfrac{1}{l-1} \ \tfrac{1}{\delta+1} \cdot a_1^1 \\
& + \ g_\delta^1 \\
& + \ \tfrac{\delta}{\delta+1} g_{\delta+1}^1 + \tfrac{\delta}{\delta+2} g_{\delta+2}^1 + \cdots + \tfrac{\delta}{\delta+l-2} g_{\delta+l-2}^1 + \tfrac{\delta}{\delta+l-1} g_{\delta+l-1}^1 .
\end{aligned} \tag{18}$$

Using (14) and (15) formula (18) implies the specific equations

$$a_1^l = \tfrac{1}{2} a_1^{l-1} + \tfrac{1}{2\cdot3} \cdot a_1^{l-2} + \tfrac{1}{3\cdot4} \cdot a_1^{l-3} + \cdots + \tfrac{1}{(l-1)l} a_1^1 + 1 \tag{19}$$

and

$$
\begin{aligned}
h_1^l &= \tfrac{1}{2}a_1^{l-1} + \tfrac{1}{2\cdot 3}\cdot a_1^{l-2} + \tfrac{1}{3\cdot 4}\cdot a_1^{l-3} + \cdots + \tfrac{1}{(l-1)l}a_1^1 + h_1^1 \\
&+ h_2^1\cdot\tfrac{1}{2}\left[\tfrac{1}{2}a_1^{l-2} + \tfrac{1}{2\cdot 3}a_1^{l-3} + \tfrac{1}{3\cdot 4}a_1^{l-4} + \cdots + \tfrac{1}{(l-2)(l-1)}a_1^1\right] + \tfrac{1}{2}h_2^1 \\
&+ \\
&\vdots \\
&+ h_{l-2}^1\cdot\tfrac{1}{l-2}\left[\tfrac{1}{2}a_1^2 + \tfrac{1}{2\cdot 3}a_1^1\right] + h_{l-2}^1\cdot\tfrac{1}{l-2} \\
&+ h_{l-1}^1\cdot\tfrac{1}{l-1}\ \tfrac{1}{2}\cdot a_1^1 + h_{l-1}^1\cdot\tfrac{1}{l-1} \\
&+ \tfrac{1}{l}h_l^1 \\
&= a_1^l + 2\cdot\tfrac{2}{2\cdot 3}\cdot a_1^{l-1} + 2\cdot\tfrac{3}{3\cdot 4}\cdot a_1^{l-2} + \cdots + 2\cdot\tfrac{l-1}{(l-1)l}a_1^2 + 2\cdot\tfrac{l}{l(l+1)}a_1^1.
\end{aligned}
$$

Thus

$$
h_1^l = a_1^l + 2\cdot\tfrac{1}{3}\cdot a_1^{l-1} + 2\cdot\tfrac{1}{4}\cdot a_1^{l-2} + \cdots + 2\cdot\tfrac{1}{l}a_1^2 + 2\cdot\tfrac{1}{l+1}a_1^1. \qquad (20)
$$

Proof of (18) by mathematical induction on l:

Induction basic: $g_\delta^2 = \frac{1}{\delta+1}(g_1^1 + \delta g_{\delta+1}^1) + g_\delta^1 = \frac{1}{\delta+1}g_1^1 + g_\delta^1 + \frac{\delta}{\delta+1}g_{\delta+1}^1$ corresponds to (18) for $l = 2$.

Induction step: If we replace g_1^l and $g_{\delta+1}^l$ in $g_\delta^{l+1} = \frac{1}{\delta+1}g_1^l + g_\delta^1 + \frac{\delta}{\delta+1}g_{\delta+1}^l$ by means of (18) then the following equation follows

$$
\begin{aligned}
g_\delta^{l+1} &= \tfrac{1}{\delta+1}g_1^1\left[\tfrac{1}{2}a_1^{l-1} + \tfrac{1}{2}\cdot\tfrac{1}{3}a_1^{l-2} + \cdots + \tfrac{1}{l-1}\tfrac{1}{l}a_1^1\right] \\
&+ \tfrac{1}{\delta+1}g_2^1\tfrac{1}{2}\left[\tfrac{1}{2}a_1^{l-2} + \tfrac{1}{2}\cdot\tfrac{1}{3}a_1^{l-3} + \cdots + \tfrac{1}{l-2}\tfrac{1}{l-1}a_1^1\right] \\
&+ \\
&\vdots \\
&+ \tfrac{1}{\delta+1}g_{l-2}^1\tfrac{1}{l-2}\left[\tfrac{1}{2}a_1^2 + \tfrac{1}{2\cdot 3}a_1^1\right] \\
&+ \tfrac{1}{\delta+1}g_{l-1}^1\cdot\tfrac{1}{2}\cdot a_1^1 \\
&+ \tfrac{1}{\delta+1}\left[g_1^1 + \tfrac{1}{2}g_2^1 + \tfrac{1}{3}g_3^1 + \cdots + \tfrac{1}{l}g_l^1\right] \qquad + g_\delta^1 \\
&+ \tfrac{\delta}{\delta+1}g_1^1\left[\tfrac{1}{\delta+2}a_1^{l-1} + \tfrac{\delta+1}{(\delta+2)(\delta+3)}a_1^{l-2} + \cdots + \tfrac{\delta+1}{(\delta+l-1)(\delta+l)}a_1^1\right] \\
&+ \tfrac{\delta}{\delta+1}g_2^1\cdot\tfrac{1}{2}\left[\tfrac{1}{\delta+2}a_1^{l-2} + \tfrac{\delta+1}{(\delta+2)(\delta+3)}a_1^{l-3} + \cdots + \tfrac{\delta+1}{(\delta+l-2)(\delta+l-1)}a_1^1\right]
\end{aligned}
$$

$+$

$\vdots$

$+\frac{\delta}{\delta+1}\frac{1}{g^1_{l-2}}\frac{1}{l-2}\left[\frac{1}{\delta+2}a^2_1+\frac{\delta+1}{(\delta+2)(\delta+3)}a^1_1\right]$

$+\frac{\delta}{\delta+1}g^1_{l-1}\frac{1}{l-1}\cdot\frac{1}{\delta+2}\cdot a^1_1$

$+\frac{\delta}{\delta+1}g^1_{\delta+1}$

$+\frac{\delta}{\delta+1}(\frac{\delta+1}{\delta+2}g^1_{\delta+2}+\frac{\delta+1}{\delta+3}g^1_{\delta+3}+\cdots+\frac{\delta+1}{\delta+l}g^1_{\delta+l})$.

If we reorganize the sum and use such partial sums as

$(\frac{1}{\delta+1}g^1_1\left[\frac{1}{2}a^{l-1}_1+\frac{1}{2}\cdot\frac{1}{3}a^{l-2}_1+\cdots+\frac{1}{l-1}\frac{1}{l}a^1_1\right])+(\frac{1}{\delta+1}g^1_1)=\frac{1}{\delta+1}g^1_1a^l_1$

$(\frac{1}{\delta+1}g^1_2\frac{1}{2}\left[\frac{1}{2}a^{l-2}_1+\frac{1}{2}\cdot\frac{1}{3}a^{l-3}_1+\cdots+\frac{1}{l-2}\frac{1}{l-1}a^1_1\right])+(\frac{1}{\delta+1}\frac{1}{2}g^1_2)=\frac{1}{\delta+1}g^1_2\frac{1}{2}a^{l-1}_1$

(and so on) then it follows that (18) is valid for $l+1$.

Now, we will show the following properties of the sequence $(a^l)_{l=1,2,\ldots}$
(where $a^l := a^l_1$ for $l=1,2,\cdots$):

$$(i) \qquad (a^l)_{l=1,2,\ldots} \text{ is strictly increasing.} \tag{21}$$

This property can be proved by a simple mathematical induction.
Clearly $a^1=1<a^2=\frac{3}{2}$ according to (19).

$a^l=\frac{1}{2}a^{l-1}+\frac{1}{2}\cdot\frac{1}{3}a^{l-2}+\cdots+\frac{1}{l-1}\frac{1}{l}a^1+1<$

$a^{l+1}=\frac{1}{2}a^l+\frac{1}{2}\cdot\frac{1}{3}a^{l-1}+\cdots+\frac{1}{l-1}\frac{1}{l}a^2+\frac{1}{l}\frac{1}{l+1}a^1+1$

follows from $a^{l-1}<a^l,\cdots,a^2<a^3,a^1<a^2$.

$$(ii) \qquad \text{The sequence } (a^l-a^{l-1})_{l=1,2,\ldots} \text{ is bounded.} \tag{22}$$

Firstly $a^{l+1}=\frac{1}{2}a^l+\frac{1}{2}\cdot\frac{1}{3}a^{l-1}+\frac{1}{3}\cdot\frac{1}{4}a^{l-2}+\cdots+\frac{1}{l-1}\frac{1}{l}a^2+\frac{1}{l}\frac{1}{l+1}a^1+1$

$=\frac{1}{2}a^l+\frac{1}{2}a^{l-1}-\frac{1}{3}(a^{l-1}-a^{l-2})-\cdots-\frac{1}{l}(a^2-a^1)-\frac{1}{l+1}a^1+1.$

Since $a^{l+1}>a^l$ it is necessary that $\frac{1}{3}(a^{l-1}-a^{l-2})<1$ and thus $a^{l-1}-a^{l-2}<3$.

For $l+1=3,4,5,\cdots$ we get $a^2-a^1<3,\ a^3-a^2<3,\ a^4-a^3<3,\cdots$.

$$(iii) \qquad \lim_{l\to\infty}\frac{a^l}{a^{l-1}}=1 \tag{23}$$

follows from (21) and (22) in both cases that $\lim_{l\to\infty}a^l$ exists or that $\lim_{l\to\infty}a^l=\infty$.

Finally we will show that $\lim\limits_{l\to\infty} \frac{h^l}{a^l} = \infty$.

According to (20) $h^{l+1} = a^{l+1} + \frac{2}{3}a^l + \frac{2}{4}a^{l-1} + \cdots + \frac{2}{l+2}a^1$.

Thus, $\frac{h^{l+1}}{a^{l+1}} = 1 + \frac{2}{3}\frac{a^l}{a^{l+1}} + \frac{2}{4}\frac{a^{l-1}}{a^{l+1}} + \cdots + \frac{2}{l+2}\frac{a^1}{a^{l+1}}$

where

$$\frac{a^j}{a^{l+1}} = \frac{a^j}{a^{j+1}} \cdot \frac{a^{j+1}}{a^{j+2}} \cdot \ \cdots \ \cdot \frac{a^l}{a^{l+1}} \text{ (for } j < l). \tag{24}$$

Let $l+1-L$ be the number of terms $\frac{a^j}{a^{l+1}} \geq \frac{1}{3}$ $(l \geq j \geq L)$. We will show that $l+1-L$ tends to infinity if l tends to infinity.

Then $\frac{h^{l+1}}{a^{l+1}} \to \infty$ follows since these quotients are related to Harmonic series.

The properties (21) and (23) imply that

$\forall\, \varepsilon > 0 \; \exists\, L(\varepsilon): \; 1 \geq \frac{a^i}{a^{i+1}} \geq 1 - \varepsilon \; \forall\, i \geq L(\varepsilon)$.

Using (24) we obtain that $\frac{a^j}{a^{l+1}} \geq (1-\varepsilon)^{l+1-j} \; \forall\, l \geq j \geq L(\varepsilon)$.

$\frac{a^j}{a^{l+1}} \geq \frac{a^{L(\varepsilon)}}{a^{l+1}} \geq (1-\varepsilon)^{l+1-L(\varepsilon)} \geq \frac{1}{3}$ is valid if and only if

$l + 1 - L(\varepsilon) \leq \frac{-ln3}{ln(1-\varepsilon)}$.

If ε tends to 0 then $\frac{-ln3}{ln(1-\varepsilon)}$ tends to infinity

and also the number of terms $\frac{a^j}{a^{l+1}} \geq \frac{1}{3}$ $(l \geq j \geq L(\varepsilon))$. ■

B Completion of Example 2

Following configurations $C_a(l), \cdots, C_i(l)$ of the online and offline servers can occur if the corresponding requests r given by the adversary and the answers by the adversary are as below:

$C_a(l):$ ON is located on $p_i, p_j, \; i,j \in \{1,2,3\}, \; d_0, d_0+d_1$.

OFF is located on $p_i, p_j, \; d_0, d_0+d_1$.

$r:$ one server on $p_q, \; q \in \{1,2,3\} \setminus \{i,j\}$

answer by the adversary: the offline server on $\begin{cases} p_{q-1} & \text{if } q \in \{2,3\} \\ p_3 & \text{if } q = 1 \end{cases}$ is moved.

($C_a(l)$ represents the initial configuration for $l = 1, i = 1$ and $j = 2$.)

$C_b(l):$ ON is located on $p_i, p_j, \; i,j \in \{1,2,3\}, \; d_0, d_0+d_1$.

OFF is located on $p_i, p_q, \; q \in \{1,2,3\} \setminus \{i,j\}, \; d_0, d_0+d_1$.

$r:$ one server on p_q; no server is moved by the adversary.

$C_c(l):$ ON is located on $p_1, p_2, p_3, \; d_0+d_1$.

OFF is located on $p_i, p_j,\ i, j \in \{1, 2, 3\},\ d_0, d_0 + d_1$.

$r:$ one server on $d_0 + 1$ and one server on $d_0 + d_1 + d_2$

answer by the adversary: the offline servers on $d_0, d_0 + d_1$ are moved.

$C_d(l):$ ON is located on $p_1, p_2, p_3,\ d_0$.

OFF is located on $p_i, p_j,\ i, j \in \{1, 2, 3\},\ d_0, d_0 + d_1$.

$r:$ one server on $d_0 + d_1$; no server is moved by the adversary.

$C_e(l):$ ON is located on $p_i,\ i \in \{1, 2, 3\},\ d_0, d_0 + d_1, d_0 + d_1 - d_3$.

OFF is located on $p_j, p_q,\ j, q \in \{1, 2, 3\} \setminus \{i\},\ d_0, d_0 + d_1$.

$r:$ one server on p_j and one server on p_q; no server is moved by the adversary.

$C_f(l):$ ON is located on $p_i,\ i \in \{1, 2, 3\},\ d_0, d_0 + d_1, d_0 + d_1 - d_3$.

OFF is located on $p_i, p_j,\ j \in \{1, 2, 3\} \setminus \{i\},\ d_0, d_0 + d_1$.

$r:$ one server on p_j; no server is moved by the adversary.

$C_g(l):$ ON: is located on $p_i, p_j,\ i, j \in \{1, 2, 3\},\ d_0 + d_1, d_0 + d_1 - d_3$

OFF: is located on $p_i, p_j,\ d_0, d_0 + d_1$

$r:$ one server on d_0; no server is moved by the adversary.

$C_h(l):$ ON is located on $p_i, p_j,\ i, j \in \{1, 2, 3\},\ d_0, d_0 + d_1 - d_3$.

OFF is located on $p_i, p_j,\ d_0, d_0 + d_1$.

$r:$ one server on $d_0 + d_1$; no server is moved by the adversary.

$C_i(l):$ ON is located on $p_1, p_2, p_3,\ d_0 + d_1 - d_3$.

OFF is located on $p_i, p_j,\ i, j \in \{1, 2, 3\},\ d_0, d_0 + d_1$.

r: one server on d_0 and one server on $d_0 + d_1$; no server is moved by the adversary.

C Computations relating to Example 3

If we want to compute absorbing probabilities $P_C(s'^{(i)}|s, r)$ we need for the corresponding Markov chains besides the states $s =: s^{(0)}$ and the absorbing states $s'^{(1)}, s'^{(2)}, \cdots, s'^{(6)}$ also the states

$s^{(1)} = (1, 0, 1, 1, 1, 0)^T, s^{(2)} = (1, 0, 1, 1, 0, 1)^T, s^{(3)} = (1, 0, 1, 0, 1, 1)^T,$

$s^{(4)} = (1, 0, 0, 1, 1, 1)^T, s^{(5)} = (0, 1, 1, 1, 1, 0)^T, s^{(6)} = (0, 1, 1, 1, 0, 1)^T,$

$s^{(7)} = (0,1,1,0,1,1)^T, s^{(8)} = (0,1,0,1,1,1)^T$
(which are transient states).

For example $(s^{(0)}, s^{(2)}, s^{(6)}, s^{(2)}, s^{(6)}, s'^4)$ is a realization of a time-homogeneous Markov chain with the absorbing state s'^4, where the corresponding surrogate request sequence $(\bar{r}^j)_{j=1,2,\cdots}$ is
$((1,0,0,0,0,0)^T, (0,1,0,0,0,0)^T, (1,0,0,0,0,0)^T, (0,1,0,0,0,0)^T, (1,0,0,0,0,0)^T)$
as described in Section 4.

If we want to compute the absorbing probabilities $P_C(s'^{(i)}|s,r)$ we need the (one-step) transition probabilities.

At first we give the matrix of transition probabilities from the transient states $s^{(0)}, s^{(1)}, \cdots, s^{(8)}$ into these transient states:

$$B = \begin{pmatrix} 0 & \frac{\frac{1}{d_{61}}}{N_f} & \frac{\frac{1}{d_{51}}}{N_f} & \frac{\frac{1}{d_{41}}}{N_f} & \frac{\frac{1}{d_{31}}}{N_f} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\frac{1}{d_{12}}}{N_f^4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\frac{1}{d_{12}}}{N_f^3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\frac{1}{d_{12}}}{N_f^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\frac{1}{d_{12}}}{N_f^1} \\ 0 & \frac{\frac{1}{d_{12}}}{N_f^8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\frac{1}{d_{12}}}{N_f^7} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\frac{1}{d_{12}}}{N_f^6} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\frac{1}{d_{12}}}{N_f^5} & 0 & 0 & 0 & 0 \end{pmatrix}$$

where $d_{ij} := d(p_i, p_j)$, $N_f = \frac{1}{d_{61}} + \frac{1}{d_{51}} + \frac{1}{d_{41}} + \frac{1}{d_{31}}$

$N_f^1 = \frac{1}{d_{12}} + \frac{1}{d_{62}} + \frac{1}{d_{52}} + \frac{1}{d_{42}}$, $N_f^5 = \frac{1}{d_{21}} + \frac{1}{d_{61}} + \frac{1}{d_{51}} + \frac{1}{d_{41}}$

$N_f^2 = \frac{1}{d_{12}} + \frac{1}{d_{62}} + \frac{1}{d_{52}} + \frac{1}{d_{32}}$, $N_f^6 = \frac{1}{d_{21}} + \frac{1}{d_{61}} + \frac{1}{d_{51}} + \frac{1}{d_{31}}$

$N_f^3 = \frac{1}{d_{12}} + \frac{1}{d_{62}} + \frac{1}{d_{42}} + \frac{1}{d_{32}}$, $N_f^7 = \frac{1}{d_{21}} + \frac{1}{d_{61}} + \frac{1}{d_{41}} + \frac{1}{d_{31}}$.

$N_f^4 = \frac{1}{d_{12}} + \frac{1}{d_{52}} + \frac{1}{d_{42}} + \frac{1}{d_{32}}$, $N_f^8 = \frac{1}{d_{21}} + \frac{1}{d_{51}} + \frac{1}{d_{41}} + \frac{1}{d_{31}}$.

$\bar{B}$ is the matrix of transition probabilities from the transient states into the absorbing states:

$$
\bar{B} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\frac{1}{d_{52}}}{N_f^4} & \frac{\frac{1}{d_{42}}}{N_f^4} & \frac{\frac{1}{d_{32}}}{N_f^4} & 0 & 0 & 0 \\
\frac{\frac{1}{d_{62}}}{N_f^3} & 0 & 0 & \frac{\frac{1}{d_{42}}}{N_f^3} & \frac{\frac{1}{d_{32}}}{N_f^3} & 0 \\
0 & \frac{\frac{1}{d_{62}}}{N_f^2} & 0 & \frac{\frac{1}{d_{52}}}{N_f^2} & 0 & \frac{\frac{1}{d_{32}}}{N_f^2} \\
0 & 0 & \frac{\frac{1}{d_{62}}}{N_f^1} & 0 & \frac{\frac{1}{d_{52}}}{N_f^1} & \frac{\frac{1}{d_{42}}}{N_f^1} \\
\frac{\frac{1}{d_{51}}}{N_f^8} & \frac{\frac{1}{d_{41}}}{N_f^8} & \frac{\frac{1}{d_{31}}}{N_f^8} & 0 & 0 & 0 \\
\frac{\frac{1}{d_{61}}}{N_f^7} & 0 & 0 & \frac{\frac{1}{d_{41}}}{N_f^7} & \frac{\frac{1}{d_{31}}}{N_f^7} & 0 \\
0 & \frac{\frac{1}{d_{61}}}{N_f^6} & 0 & \frac{\frac{1}{d_{51}}}{N_f^6} & 0 & \frac{\frac{1}{d_{31}}}{N_f^6} \\
0 & 0 & \frac{\frac{1}{d_{61}}}{N_f^5} & 0 & \frac{\frac{1}{d_{51}}}{N_f^5} & \frac{\frac{1}{d_{41}}}{N_f^5}
\end{pmatrix}.
$$

Finally, the absorbing probabilities can be computed by the following linear systems (see [6], Theorem 6.6 for example)

$$
u^{(j)} = B\ u^{(j)} + \bar{B}^{(j)}
$$

with variables $u_i^{(j)}$ $(i = 0, \cdots, 8)$ and where $\bar{B}^{(j)}$ is the j-th column of matrix $\bar{B}$. The solution value of $u_i^{(j)}$ is the absorbing probability of state $s'^{(j)}$ $(j \in \{1, \cdots, 6\})$, if the initial state of the corresponding Markov chain is $s^{(i)}$ $(i \in \{0, 1, \cdots, 8\})$. Thus $u_0^{(j)} = P_C(s'^{(j)}|s, r)$.